US006233567B1

(12) United States Patent
Cohen

(10) **Patent No.: US 6,233,567 B1**
(45) **Date of Patent: *May 15, 2001**

(54) **METHOD AND APPARATUS FOR SOFTWARE LICENSING ELECTRONICALLY DISTRIBUTED PROGRAMS**

(75) Inventor: **Aaron Michael Cohen**, Beaverton, OR (US)

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **08/920,679**

(22) Filed: **Aug. 29, 1997**

(51) **Int. Cl.^{7}** **H04L 9/00**

(52) **U.S. Cl.** **705/59**

(58) **Field of Search** 380/21, 23, 49, 380/30; 705/54, 51, 59, 52

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,222,134	*	6/1993	Waite et al.	705/59
5,337,357	*	8/1994	Chou et al.	380/4
5,490,216	*	2/1996	Richardson, III	705/59
5,553,143	*	9/1996	Ross et al.	705/59
5,555,304	*	9/1996	Hasebe et al.	380/4
5,586,186	*	12/1996	Yuval et al.	380/30
5,638,443	*	6/1997	Stefik et al.	705/54
5,651,064	*	7/1997	Newell	380/4
5,724,425	*	3/1998	Chang et al.	705/52
5,737,419	*	4/1998	Ganesan	380/21
5,758,068	*	5/1998	Brandt et al.	395/186
5,926,549	*	7/1999	Pinkas	713/168

FOREIGN PATENT DOCUMENTS

0795809 * 9/1997 (EP) .

OTHER PUBLICATIONS

Albert et al; "Combatting Software Piracy by Encryption and Key Management", Apr. 1984; Computer; pp. 68–73.*

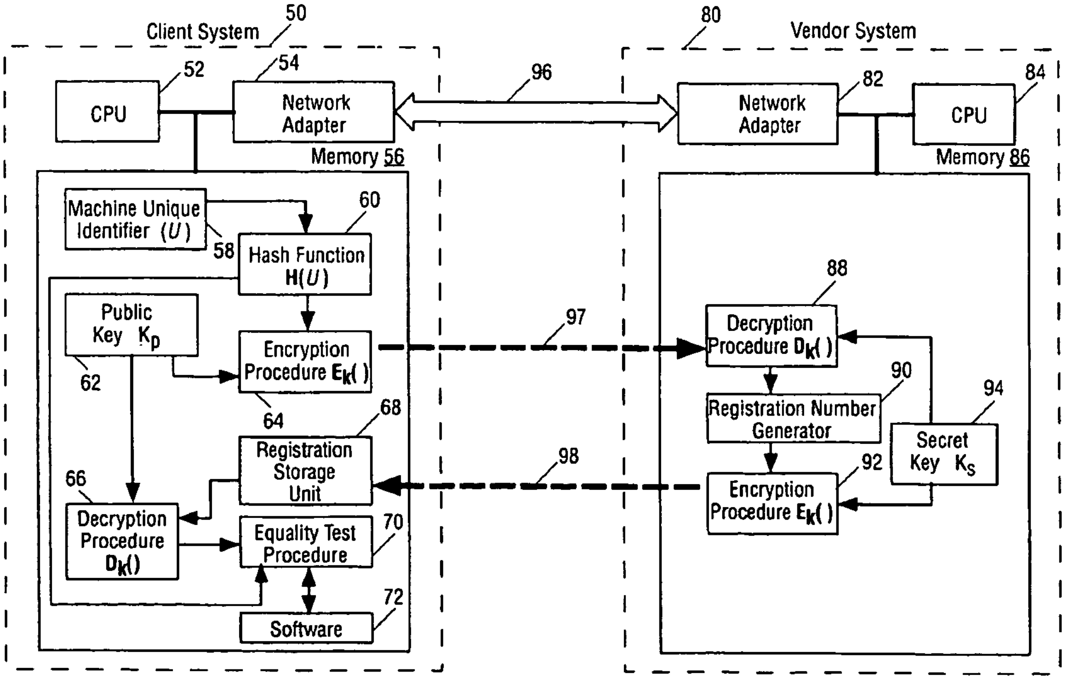
* cited by examiner

Primary Examiner—Tod Swann
Assistant Examiner—Matthew Smithers
(74) *Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

(57) **ABSTRACT**

A method including the steps of receiving a registration identifier for a client; generating a registration key based on the registration identifier; and transmitting the registration key to the client.

21 Claims, 4 Drawing Sheets



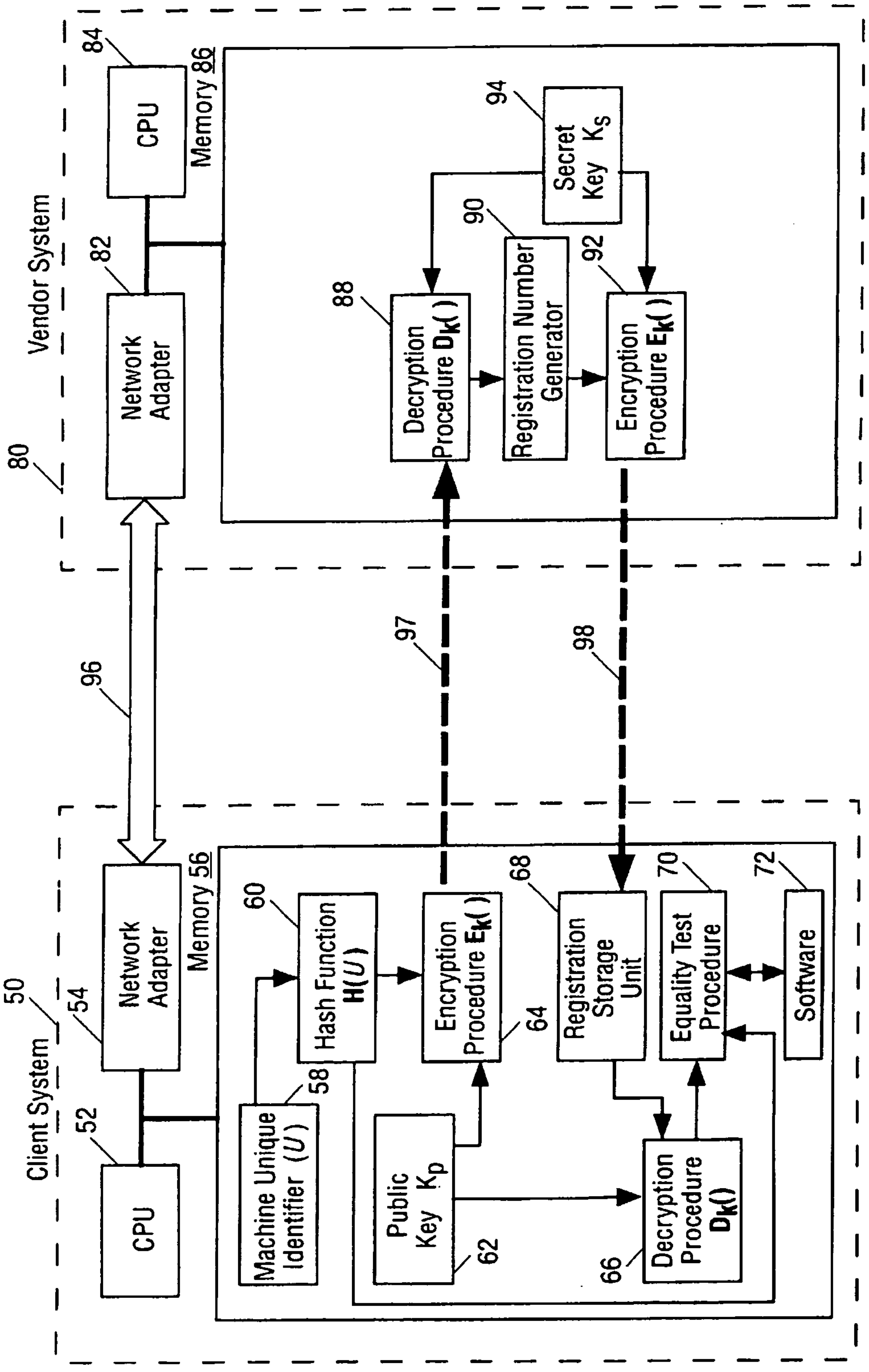

Vendor System
80
CPU
84
Memory 86
Network Adapter
82
Secret Key K_S
94
Decryption Procedure $D_K(\ )$
88
Registration Number Generator
90
Encryption Procedure $E_K(\ )$
92
96
97
98
Client System
50
CPU
52
Network Adapter
54
Memory 56
Machine Unique Identifier (U)
58
Hash Function $H(U)$
60
Encryption Procedure $E_K(\ )$
64
Public Key K_p
62
Registration Storage Unit
68
Decryption Procedure $D_K()$
66
Equality Test Procedure
70
Software
72

FIG. 1

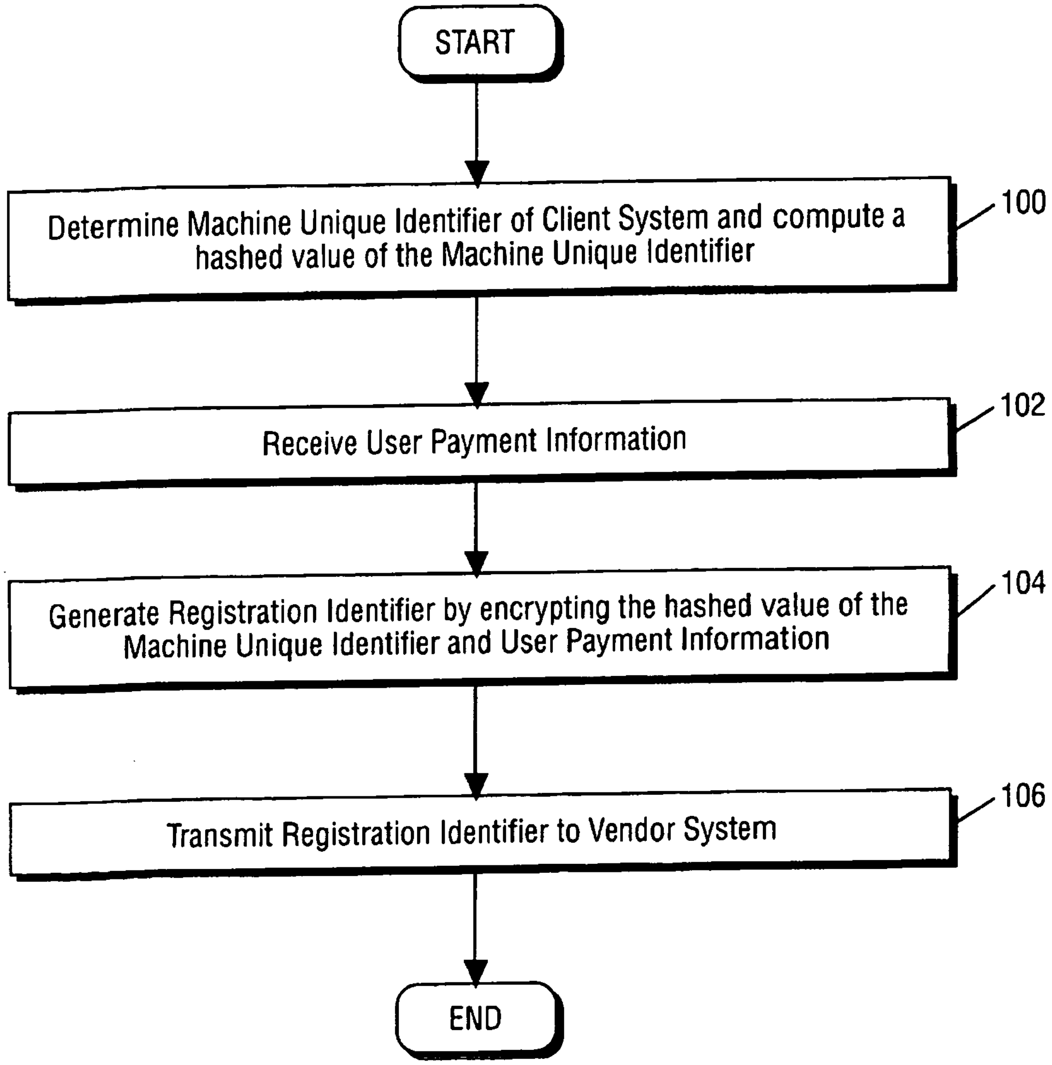

START
Determine Machine Unique Identifier of Client System and compute a hashed value of the Machine Unique Identifier
100
Receive User Payment Information
102
Generate Registration Identifier by encrypting the hashed value of the Machine Unique Identifier and User Payment Information
104
Transmit Registration Identifier to Vendor System
106
END

FIG. 2

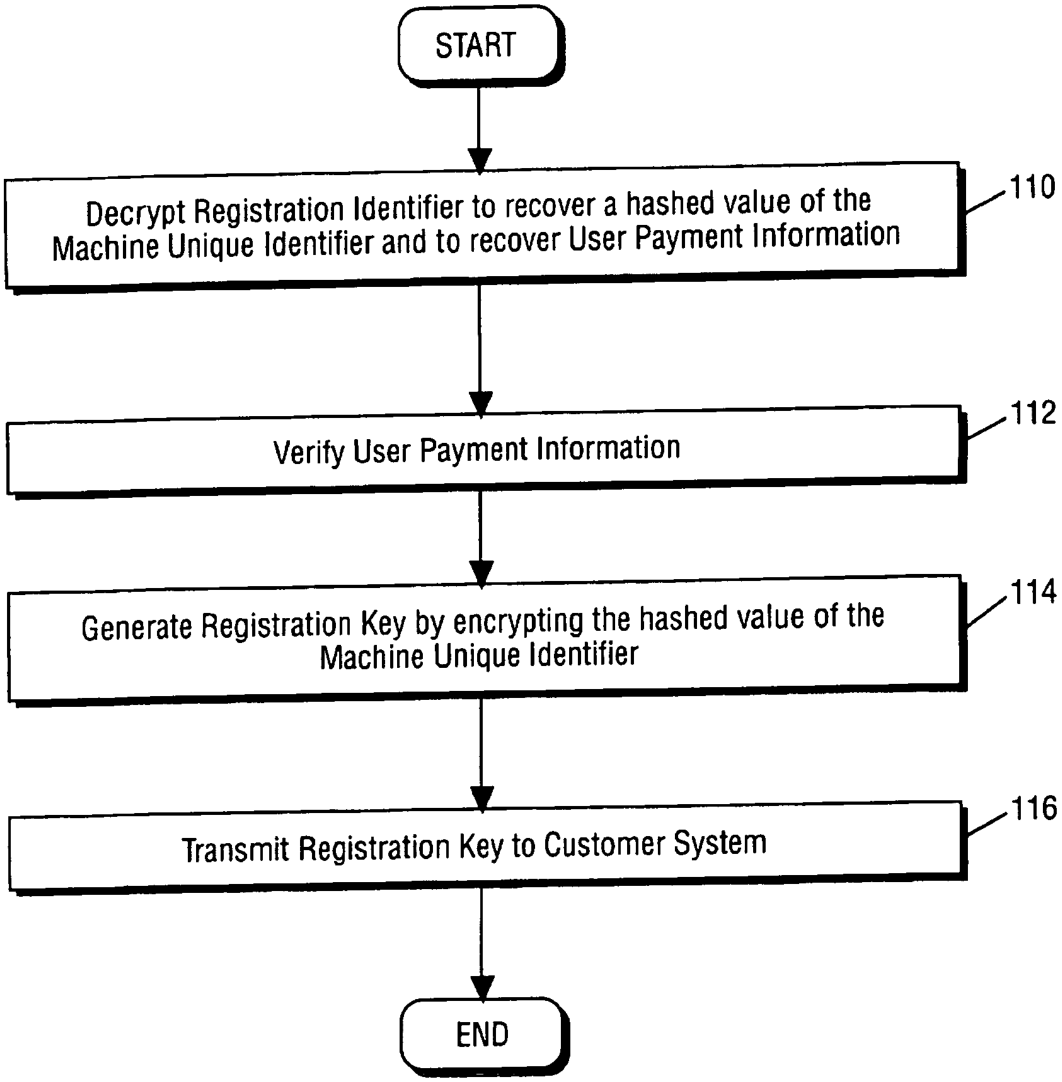

START
Decrypt Registration Identifier to recover a hashed value of the Machine Unique Identifier and to recover User Payment Information
110
Verify User Payment Information
112
Generate Registration Key by encrypting the hashed value of the Machine Unique Identifier
114
Transmit Registration Key to Customer System
116
END

FIG. 3

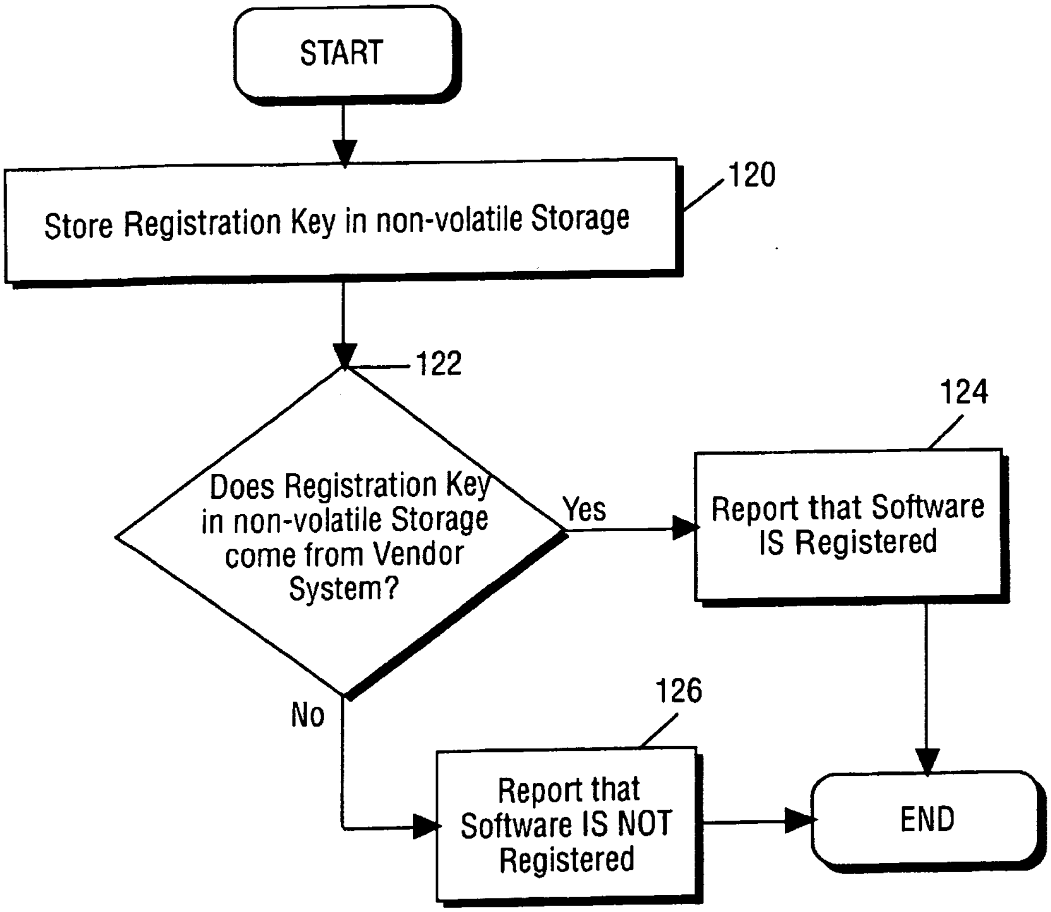

START
Store Registration Key in non-volatile Storage
120
122
Does Registration Key in non-volatile Storage come from Vendor System?
Yes
No
124
Report that Software IS Registered
126
Report that Software IS NOT Registered
END

FIG. 4

METHOD AND APPARATUS FOR SOFTWARE LICENSING ELECTRONICALLY DISTRIBUTED PROGRAMS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of use of computer software registration. More particularly, the present invention relates to secure registration of computer software.

2. Description of Related Art

The use of wide-area-networks such as the Internet to distribute software has become a very popular way to distribute software. The software can be programmed to be—until a license is purchased—either fully functional for a "trial period" of a certain duration, or partially functional. Providing potential customers the ability to download functional versions of a particular software allows access to an audience base that is limited only by the means of distribution (e.g., the size of the audience which has access to the Internet).

In addition, using networks to distribute demonstration or "demo" software is cost effective for the software company, as the company does not need to first place the demo software onto a distribution medium such as floppy disks or compact disk read-only-memory (CD-ROM) disks. Moreover, the company does not have to create or pay for packaging, nor maintain an inventory. The cost saving is especially beneficial in helping companies save marketing funds, which can be invested in other programs.

However, these cost savings disappear when the company has to ensure that customers who download the software pay for the software. Companies which put functionally limited versions of their software on the network requires a customer to send in payment for the software before the customer is sent a fully functional version. These companies must maintain a stock of packaged software, exactly the problem that a network-based distribution method attempts to solve.

Companies which put a time limit or other restrictions on their software require the customer to pay for a license before the customer is sent a "key code". The key code is entered into the program, which then unlocks any restrictions. The problem associated with this scheme is that the same key code can be used for any copy of the software, so multiple individuals can unlock their respective copies of the software by simply purchasing one license and distributing the received key code amongst themselves.

Thus, it would be preferrable to have a software distribution scheme that overcomes the problems associated with these methods.

SUMMARY OF THE INVENTION

A method including the steps of receiving a registration identifier for a client; generating a registration key based on the registration identifier; and transmitting the registration key to the client.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a client system and a vendor system configured in accordance to a preferred embodiment of the present invention.

FIG. 2 is a flow diagram of the operation of the client system for initiating a request for registration of a software license.

FIG. 3 is a flow diagram of the operation of the server system in receiving the request of the client system and generating a registration key for the software on the client system.

FIG. 4 is a flow diagram of the operation of the client system after it has received the registration key for the software.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method and apparatus for distributing software licenses. For purposes of explanation, specific embodiments are set forth to provide a thorough understanding of the present invention. However, it will be understood by one skilled in the art, from reading this disclosure, that the invention may be practiced without these details. Further, although the present invention is described through the use of software distribution over the Internet, most, if not all, aspects of the invention apply to software distribution in general. Moreover, well-known elements, devices, process steps and the like are not set forth in detail in order to avoid obscuring the present invention.

Through the use of public key cryptography, one-way hash functions and unique machine identification, software registration is provided which is individualized to a particular computer. Thus, software registration is "locked-in" to a particular computer and cannot be used on another computer—preventing the sharing of key codes.

In order to describe this system of software distribution, explanation is first provided below for public key cryptography, one-way hash functions and unique machine identification.

Public Key Cryptography

Public key cryptography provides the ability for two parties to send information securely between themselves. Unlike symmetrical cryptography, which requires a shared secret key, public key cryptography uses one key, a "public" key, to encode information and another complementary key, a "private key" to decode encrypted information. The security of the system lies in the method used to create the key pair and the belief that it is very difficult to determine the private key from the public key.

In use, a user publishes the public key and keeps the private key secret. Parties wishing to send a message to the user encrypt the message with the user's published public key and send it to the user. Upon receiving the encrypted message, the user decrypts the message with the user's private key, thereby recovering the original message.

The user can also "sign" a document by using the user's private key. The user would encrypt the message with the private key, and other parties would decrypt the message with the user's public key. Only documents encrypted with the user's private key will be intelligible when decrypted with the user's public key.

Mathematically, encryption is represented by:

$$C=E_{k1}(M)$$

and decryption is:

$$M=D_{k2}(C),$$

where M is the original message, C is the encrypted message, k1 is the public key, k2 is the private key, E() is the encryption function and D() is the decryption function. For signing of documents, the keys used would be reversed.

One-Way Hash Functions

A one-way hash function is a function which cannot be easily reversed. Specifically, given an input, an output is easy to generate, but given the output, the input is practically impossible to reconstruct. Also, given an output, it would be very difficult to generate input data which hashes to the output value. One-way hash functions can output more, less, or the same amount of information (e.g., number of bits) from a given input. To be useful, the hash function should return practically unique values for a given input. Usually, the hash values have less information than the input data.

One-way hash functions are useful in constructing "signatures" of documents. For example, if user A has a document, and user B wants to prove to user A that he has the same document, user B can run an agreed upon one-way hash function and send the result to user A, who can run the same one-way hash function and compare hash values. If they match, user A has strong evidence that user B has a copy of the same document.

Mathematically, the hash function is represented by:

$$S=H(M)$$

where M is the original message, H() is the one-way hash function, and S is the signature of the message.

Unique Machine Identification

Modern operating systems support remote procedure calls (RPC), which requires a unique method of identifying each machine on a network. Thus, most operating systems include a way of generating universal unique identifiers (UUID), which are unique in time and space. These UUID's have a well defined layout and have preallocated portions for location information, time information, and user defined information. Every UUID created on a particular machine will have the same values for the location bits. Therefore, these bits can be used to uniquely identify a particular machine.

Software Registration

FIG. 1 is a block diagram of a client system **50** and a vendor system **80** configured in accordance with one embodiment of the present invention.

Client **50** contains a CPU **52**, which is a general purpose processor, coupled to a network adapter **54**. Also coupled to network adapter **54** and CPU **52** is a memory **56**, which stores the data and procedures which CPU **52** uses to operate.

Memory **56** of client system **50** contains a machine unique identifier U **58**; a hash function H(U) **60**; a public key Kp **62**; an encryption procedure $E_k()$ **64**; a decryption procedure $D_k()$ **66**; a registration storage unit **68**; an equality test procedure **70** and software **72**.

As discussed above, machine unique identifier U **58** is a number that is unique to client system **50**, and the size of machine unique identifier U **58** can be of any length, as generated by client system **50**.

Also, as discussed above, encryption procedure $E_k()$ **64** decryption procedure $D_k()$ **66** are used to encrypt and decrypt, respectively, messages which are received from vendor system **80**.

Public key K_p is used with encryption procedure $E_k()$ **64** to create an encrypted version of a one-way hashed machine unique identifier U **58**, as described below. Public Key K_p **62** is also used with decryption procedure $D_k()$ **66** to authenticate any registration keys received from vendor system **80**.

Registration storage unit **68** is used to store the registration key received from vendor system **80** for enabling features of software **72**.

Equality test procedure **70** is used to verify that the decrypted version of the registration key stored in registration storage unit **68** is equivalent to the one-way hashed value of machine unique identifier (U) **58**, as generated by hash function H(U) **60**. Equality test procedure **70** is interfaced with software **72** to enable/disable functionality of software **72** based on the output of equality test procedure **70**, as discussed below.

Continuing to refer to FIG. 1, vendor system **80** contains a network adapter **82** which is used to communicate with network adapter **54** of client system **50** through a network **96**. Network **96** can be a general purpose network such as the Internet or a local-area-network containing two or more systems.

Vendor system **80** also contains a CPU **84**, which can be a general purpose processor, coupled to network adapter **82**. It is to be noted that CPU **84** and CPU **52** of client system **50** can also be custom integrated circuits.

Also coupled to network adapter **82** and CPU **84** is a memory **86**. Memory **86** of vendor system and memory **56** of client system **50** can also be general purpose data storage devices or custom data storage devices such as integrated circuits and can be built into CPU **84** of vendor system **80** and CPU **52** of client system **50**, respectively.

Memory **86** of vendor system **80** contains a decryption procedure $D_k()$ **88**; a registration number generator **90**; and encryption procedure $E_k()$ **92**, and secret key K_s **94**.

Decryption procedure $D_k()$ **88** is functionally equivalent to decryption procedure $D_k()$ **66** of client system **50**. Similarly, encryption procedure $E_k()$ **92** is functionally similar to encryption procedure $E_k()$ **64** of client system **50**. However, vendor system **80** will use secret key K_s **94** and decryption procedure $D_k()$ **88** to decrypt the messages generated by encryption procedure $E_k()$ **64** of client system **50** (client system **50** using public key K_p). Also, vendor system **80** will use secret key K_s **94** in encryption procedure $E_k()$ **92** to authenticate messages which are sent to decryption procedure $D_k()$ **66** of client system **50**.

Registration number generator **90** is used to verify user payment information CC which is received from client system **50**. After payment is made, registration number generator will allow vendor system **80** to generate a registration key.

A first dotted line **97** is used to logically represent the sending of data from encryption procedure $E_k()$ **64** of client system **50** to decryption procedure $D_k()$ **88** of vendor system **80**, while a second dotted line **98** is used to logically represent the sending of encryption procedure $E_k()$ **92** of vendor system **80** to registration storage unit **68** of client system **50**. The actual data is sent over network **96** through the use of network adapter **54** and network adapter **82**.

It is to be noted that although software **70** is shown to be a separate functional block in FIG. **1**, in alternate embodiments, software **70** contains any combination of the functional and storage elements contained in memory **56** of client system **50**.

FIG. 2 is a flow diagram of the operation of the software registration system, as shown in FIG. 1, where the user decides to register the product.

In block **100**, client system **50** first determines machine unique identifier U **58**. As noted above, machine unique identifier U **58** is used to uniquely identify client system **50** and is generated by using a built-in function of the operating system. After machine unique identifier U **58** is determined, client system **50** creates a one-way hashed version of machine unique identifier U using hash function H(U) **60**. The generation of the one-way hashed version of the machine unique identifier in block **100** provides a practically unique registration code which does not allow the vendor access to any sensitive machine information, such as the network card ID number. The use of one-way hash function

H(U) allows the registration identifier to be a fixed size, independent of how many bits of information are available in machine unique identifier U **58**. A fixed size registration identified is useful for multi-platform products as each type of platform may have a different number of location specific bits in the UUID.

In block **102**, client system **50** receives user payment and other transaction specific information (CC). This is information appended to the one-way hashed version of the machine unique identifier and is whatever transaction specific information the vendor requires, such as the user's name and credit card number.

In block **104**, client system **50** generates a registration identifier R by using this formula:

$$M=H(U)+CC$$

$$R=E_{kp}(M)$$

where H( ) is hash function **60**; U is machine unique identifier **58**; CC is private, transaction specific information, such as the user's name and credit card number; M is the one-way hashed machine unique identifier with private user data appended (i.e., the "message"); E_k( ) is encryption procedure **64**; k_p is the published, public key; **62** and R is the generated registration identifier. As this information is encrypted using the published public key (Kp), it can only be decrypted and read by vendor system **80** with the private key (K_s).

In block **106**, the registration identifier (R) is transmitted to vendor system **80**. This can be done automatically by software **72**, which is contained on client system **50** over the Internet, or a text representation of the registration identifier (R) can be generated and sent to the vendor to be processed on vendor system **80**.

As described, the information contained in registration identifier (R) is encrypted before it is transmitted, so it can be transmitted using any method, either securely or non-securely.

FIG. **3** illustrates the operation of vendor system **80** where the client system **50** has transmitted registration identifier (R).

In block **110**, vendor system **80**, upon receiving the registration identifier (R), computes:

$$M=D_{ks}(R)$$

$$H(U)+CC=M$$

where R is the registration code; M is the one-way hashed machine identifier with private user data appended, recovered by decrypting R (this is split into two parts to recover H(U) and CC); D_k( ) is decryption procedure **88**; and k_s is the secret, private key **94**.

In block **112**, the private user information (CC), is used to verify payment for the software. This verification can be as simple as processing a credit card transaction or verifying that the user has sent in payment.

In block **114**, after payment is received, vendor system **80** will generate:

$$T=E_{ks}(H(U))$$

where E_k( ) is encryption procedure **92** and T is the generated registration key.

It is to be noted that as registration key (T) is based on a machine identifier which is unique for client system **50**, even if registration key (T) is compromised, it could not be used for another machine.

In block **116**, vendor system **80** will transmit registration key (T) to client system **50**. As stated above, as registration key (T) is an encrypted value of the one-way hashed value of machine unique identifier U **58**, registration key (T) can be transmitted using any means, whether it is secure or unsecure.

Further, as registration key (T) is specific for client system **50** and cannot be used by another system, the security of the key system can be compromised and the protection provided by the system would still remain.

FIG. **4** illustrates the operation of client system **50** after client system **50** has received registration key (T).

In block **120**, client system **50** will store registration key (T) in registration storage unit **68** so that it can be accessed when needed. When software **72** needs to expose or hide functionality based on the registration status, the current key is loaded from this location, decrypted, and checked for correctness as discussed in block **122**. Also, the next time software **72** runs or needs to decide if software **72** is a registered copy, client system **50** will go to block **122**.

In block **122**, equality test procedure **70** of client system **50** will determine if registration key (T) comes from vendor system **80** by checking to see if the following holds true:

$$D_{kp}(T)=H(U)$$

where: D_k( ) is decryption procedure **88**; k_p is the published, public key **62**; H( ) is the one-way hash function **60**; U is machine unique identifier **58**; and T is the registration key.

If the equality holds true, then operation will continue with block **124**. Otherwise, operation will continue with block **126**.

In block **124**, client system **50** will allow any functionality in software **72** that was previously disabled.

In block **126**, as client system **50** has detected that registration key (T) is not received from vendor system **80** or is not valid, any functionality of software **72** that is not accessible to non-paid users remain locked or hidden.

It is to be noted that any public key cryptography algorithm that can transmit arbitrary messages will work in the system. However, the security of the system is only as secure as the cryptography algorithm. For public key cryptography systems, security increases as more bits are added to the key (i.e., the key length is increased). In a preferred embodiment, the key length is at least 512 bits.

In order to prevent an attacker from trying to break the licensing scheme by modifying the executable code containing the check which disables functionality, several alternate embodiments are proposed.

First, all debug information should be removed from any executable before distribution. This makes it harder for the attacker to follow the flow of control which checks the registration code.

Second, multiple places in the code can check for the existence of a correct registration key (T). This makes it harder to bypass all of the registration checks.

Third, the registration checking code can be obfuscated, making it hard to follow and change. This can be made complicated enough that it would be more cost effective to just license the software legally.

Fourth, for even more security, the software itself could be encrypted with the private key and loaded, decrypted and run using a second loader program. This method would be secure against all but the dumping of the binary image of the running executable out to a disk file and reconstructing a executable program file from a binary image.

For maximum security, the operating system (OS) itself could be enhanced to only execute encrypted executable files. The OS would be shipped with the decryption key, but the encryption key would remain secret. To execute a program, it would have to be decrypted by the internal OS key. Since only the OS manufacturer would have the encryption key, only programs encrypted by the OS manufacturer could be run. Obviously, this level of security would affect the way that software could be written and used. However some usage models, such as game machines where most software comes from one manufacturer and no software is written on the executing machine itself, could use this security method.

In one alternate embodiment, time-limited licenses can be granted. Instead of simply decrypting and re-encrypting the unique machine identifier, an expiration date is added to the message. When client system **50** checks registration key (T), client system **50** also decrypts the expiration date and checks if the license has expired.

In the alternate embodiment, the following functions would be used on vendor system **80**:

$$M=Dk_s(R)$$

$$H(U)+CC=M$$

$$T=Ek_s(H(U)+V)$$

Where: R is the registration code; M is the one-way hashed machine identifier with private user data appended, recovered by decrypting R (this is split into two parts to recover H(U) and CC); V is the expiration date; $E_k(\ )$ is the encryption procedure; $D_k(\ )$ is the decryption procedure; k_s is the secret, private key; and, T is the generated registration key.

In addition, on client system **50**, the following function would be used:

$$Dk_p(T)=K,\ V$$

and a check performed to determine if the two following conditions hold true:

$$K=H(U)$$

and,

V has not expired. where: $D_k(\ )$ is the decryption procedure; k_p is the published, public key; H( ) is the one-way hash function; U is the machine unique identifier; T is the registration key; K is the machine identifier portion of the decrypted registration key; and V is the expiration date portion of the decrypted registration key.

It is to be noted that the unique identifier does not have to be hashed before being transmitted. In addition, no private information has to be appended for payment purposes. In another alternate embodiment, only the unique identifier is transmitted.

In yet another alternate embodiment, the unique identifier, U, is not machine specific but specific in another way, such as user or binary specific. A software distribution site could be set up to download executables that are identical except for an internal identifier.

Alternatively, the software could be distributed with an installation program that set the executable's internal unique ID to some time or location specific value. Each user would get an equivalent binary file that required a different registration key, but the registration process and key verification would be exactly as in the basic system. This would allow a user to install the software on multiple machines but not share the registration key with other users. If the unique identifier could be something person specific, such as a fingerprint, a voice print, or a handwriting signature, this alternate embodiment could be very attractive.

Using electronic "money", the entire process could be automated. A Web server could process the payment with the registration identified and send the registration key back to the user (all over a secure channel such as Secure Socket Layer) within a single transaction.

While the present invention has been particularly described with reference to the various figures, it should be understood that the figures are for illustration only and should not be taken as limiting the scope of the invention. Many changes and modifications may be made to the invention, by one having ordinary skill in the art, without departing from the spirit and scope of the invention.

What is claimed is:

1. A method comprising:

receiving an encrypted registration identifier for a client, said registration identifier contains an one-way hashed value of a machine unique identifier for said client, said registration identifier being encrypted using a public key;

decrypting said registration identifier using a private key that is matched to said public key to retrieve the one-way hashed value;

generating a registration key based on said registration identifier by encrypting the retrieved one-way hashed value; and

transmitting said registration key to said client.

2. The method of claim 1 wherein said registration identifier further contains user payment information.

3. The method of claim 2, further comprising decrypting said registration identifier to retrieve said user payment information.

4. The method of claim 3, further comprising verifying payment using said user payment information.

5. The method of claim 1 further comprising retrieving the one-way hashed value from said registration key by the client; and

comparing the client retrieved one-way hashed value to a client generated one-way hashed value.

6. The method of claim 1, wherein said generating further comprises encrypting the one-way hashed value along with an expiration time indicator.

7. A method comprising:

determining a machine unique identifier;

generating an one-way hashed value of said machine unique identifier;

encrypting said one-way hashed value of said machine unique identifier to generate a registration identifier using a public key of a server;

transmitting said registration identifier to said server;

receiving a registration key from the server, the registration key contains an encrypted form of the one-way hashed value retrieved by the server;

retrieving by a client the one-way hashed value from the registration key; and

comparing the client retrieved one-way hashed value to a client-generated one-hashed value.

8. A method comprising:

receiving a registration key;

storing said registration key in memory;

retrieving a one-way hashed value of a machine unique identifier from said registration key;

generating a one-way hashed value of a machine unique identifier from said client;

comparing said retrieved one-way hashed value of said machine unique identifier with said generated one-way hashed value of said machine unique identifier; and, providing a software enable signal only if said retrieved one-way hashed value of said machine unique identifier is equal to said generated one-way hashed value of said machine unique identifier.

9. The method of claim 8, further comprising:

retrieving an expiration time indicator from said registration key; and,

eliminating the provision of said software enable signal if said expiration time indicator indicates that said registration key has expired.

10. An apparatus comprising:

a processor;

a memory coupled to said processor and configured with instructions to cause said processor to:

receive an encrypted registration identifier for a client, said registration identifier contains an one-way hashed value of a machine unique identifier for said client, said registration identifier being encrypted using a public key;

decrypt said registration identifier using a private key that is matched to said public key, to retrieve the one-way hashed value;

generate a registration key based on said registration identifier by encrypting the one-way hashed value retrieved from said registration identifier; and,

transmit said registration key to said client.

11. The apparatus of claim 10, wherein said registration identifier contains an one-way hashed value of a machine unique identifier for said client.

12. The apparatus of claim 10 wherein said registration identifier further contains user payment information.

13. The apparatus of claim 12, where said memory contains further instructions configured to cause said processor to decrypt said registration identifier to retrieve said user payment information.

14. The apparatus of claim 13 where said memory contains further instructions configured to cause said processor to verify payment using said user payment information.

15. The apparatus of claim 10, where, to generate said registration key based on said registration identifier, said memory contains further instructions configured to cause said processor to encrypt the retrieved one-way hashed value along with an expiration time indicator.

16. An article of manufacture comprising:

a machine-readable medium having instructions which, when executed by a machine, cause the machine to

receive an encrypted registration identifier for a client, said registration identifier contains a one-way hashed value of a machine unique identifier for said client, said registration identifier being encrypted using a public key;

decrypt said registration identifier using a private key that is matched to said public key to retrieve the one-way hashed value;

generate a registration key based on said registration identifier by encrypting the retrieved one-way hashed value into the registration key; and

transmit said registration key to said client after verifying payment.

17. The article of manufacture of claim 16 wherein the machine-readable medium includes further instructions which cause the machine to process a user payment, based upon user payment information retrieved from the registration identifier, before transmitting the registration key.

18. The article of manufacture of claim 17 wherein the machine readable medium includes further instructions which cause the machine to include an expiration time indicator when generating the registration key.

19. An article of manufacture comprising:

a machine-readable medium having instructions which, when executed by a client machine, cause the machine to

(a) determine a machine unique identifier for said machine;

(b) generate a one-way hashed value of said machine unique identifier;

(c) encrypt said one-way hashed value to generate a registration identifier using a public key of a server;

(d) transmit said registration identifier to said server;

(e) receive a registration key from the server, the registration key contains an encrypted form of the one-way hashed value;

(f) retrieve the one-way hashed value from the registration key;

(g) determine a machine unique identifier for said machine and generate a one-way hashed value thereof; and

(h) compare the retrieved one-way hashed value in (f) to the one-way hashed value in (g).

20. The article of manufacture of claim 19 wherein the machine-readable medium includes further instructions which cause the machine to provide a software enable signal only if the comparison in (h) indicates that the one-hashed value retrieved in (f) is equal to the one generated in (g).

21. The article of manufacture of claim 20 wherein the machine-readable medium includes further instructions which cause the machine to retrieve an expiration time indicator from the registration key, and not provide the enable signal if the expiration time indicator indicates that the registration key has expired.

* * * * *

US006125349A

United States Patent [19]

Maher

[11] **Patent Number:** **6,125,349**

[45] **Date of Patent:** **Sep. 26, 2000**

[54] **METHOD AND APPARATUS USING DIGITAL CREDENTIALS AND OTHER ELECTRONIC CERTIFICATES FOR ELECTRONIC TRANSACTIONS**

[75] Inventor: **David P. Maher**, Ponte Vedra Beach, Fla.

[73] Assignee: **AT&T Corp.**, New York, N.Y.

[21] Appl. No.: **09/107,785**

[22] Filed: **Jun. 30, 1998**

Related U.S. Application Data

[60] Provisional application No. 60/060,643, Oct. 1, 1997.

[51] **Int. Cl.**^{7} **G06F 17/60**

[52] **U.S. Cl.** **705/1**; 705/37; 705/53; 705/56; 705/59; 705/75; 705/76; 705/38

[58] **Field of Search** 705/1, 26, 35, 705/37, 38, 39, 53, 36, 59, 75, 76; 380/24

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,623,547	4/1997	Jones et al. .	
5,671,279	9/1997	Elgamal .	
5,692,132	11/1997	Hogan .	
5,715,314	2/1998	Payne et al. .	
5,732,400	3/1998	Mandler et al.	705/26
5,790,677	8/1998	Fox et al.	380/24
5,797,133	8/1998	Jones et al.	705/38
5,878,403	3/1999	DeFrancesco et al.	705/38

OTHER PUBLICATIONS

"Netscape To Extend Business Extranet", Aug. 1997; vol. 15; No. 8; Multimedia Monitor; [retreived on Aug. 1999]; retrieved from Dialog Information Services.

Primary Examiner—James P. Trammell
Assistant Examiner—Chinor M. Lee

[57] **ABSTRACT**

A method for performing electronic transactions, comprising receiving a long-term certificate, authenticating a user associated with the long-term certificate, and then sending a short-term certificate to the authenticated user. In addition, risk associated with the user can be evaluated, and this risk information, as well as other information, can be included in the short-term certificate.

31 Claims, 2 Drawing Sheets

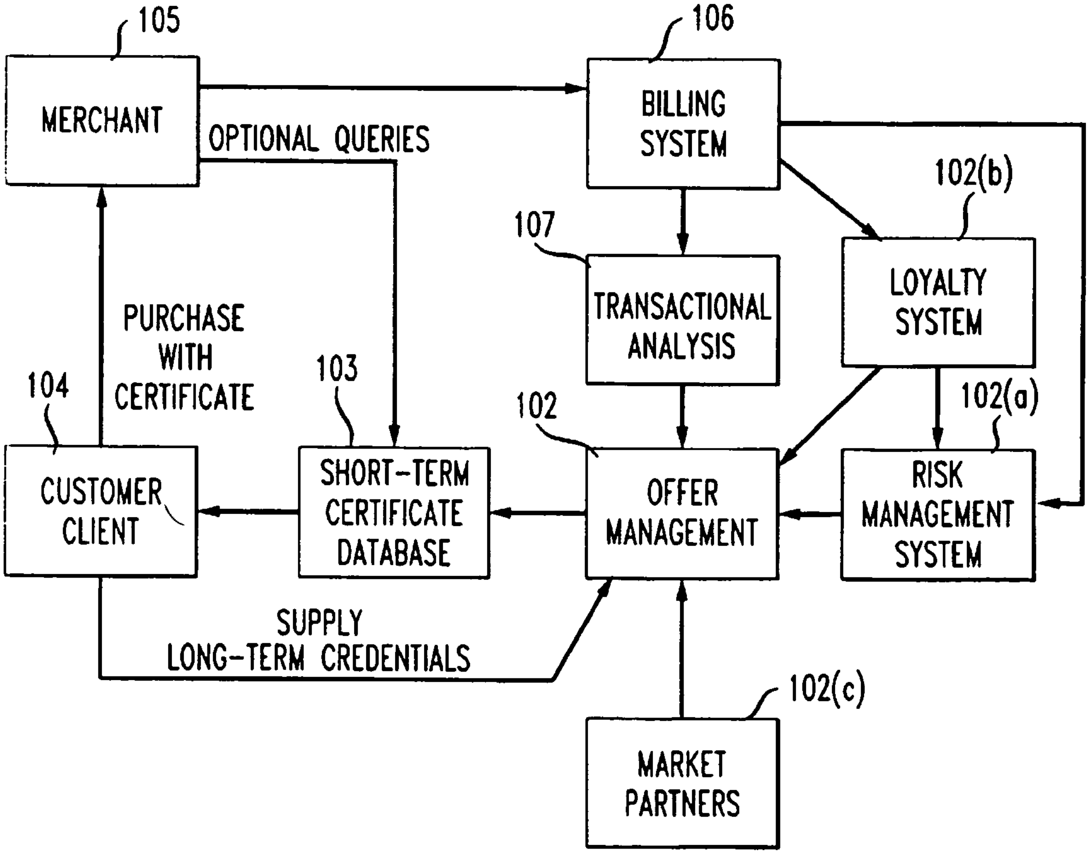


FIG. 1

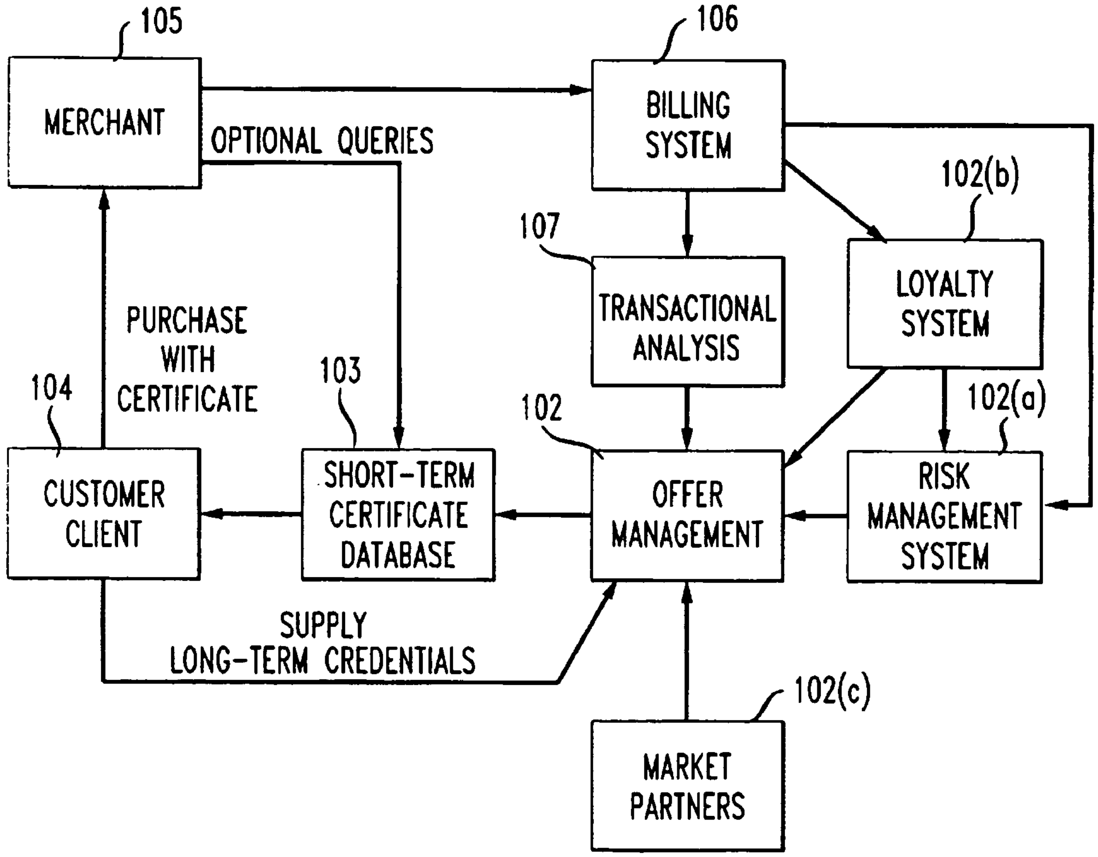

105
MERCHANT
OPTIONAL QUERIES
106
BILLING SYSTEM
107
TRANSACTIONAL ANALYSIS
102(b)
LOYALTY SYSTEM
PURCHASE WITH CERTIFICATE
104
103
102
102(a)
CUSTOMER CLIENT
SHORT-TERM CERTIFICATE DATABASE
OFFER MANAGEMENT
RISK MANAGEMENT SYSTEM
SUPPLY LONG-TERM CREDENTIALS
102(c)
MARKET PARTNERS

FIG. 2

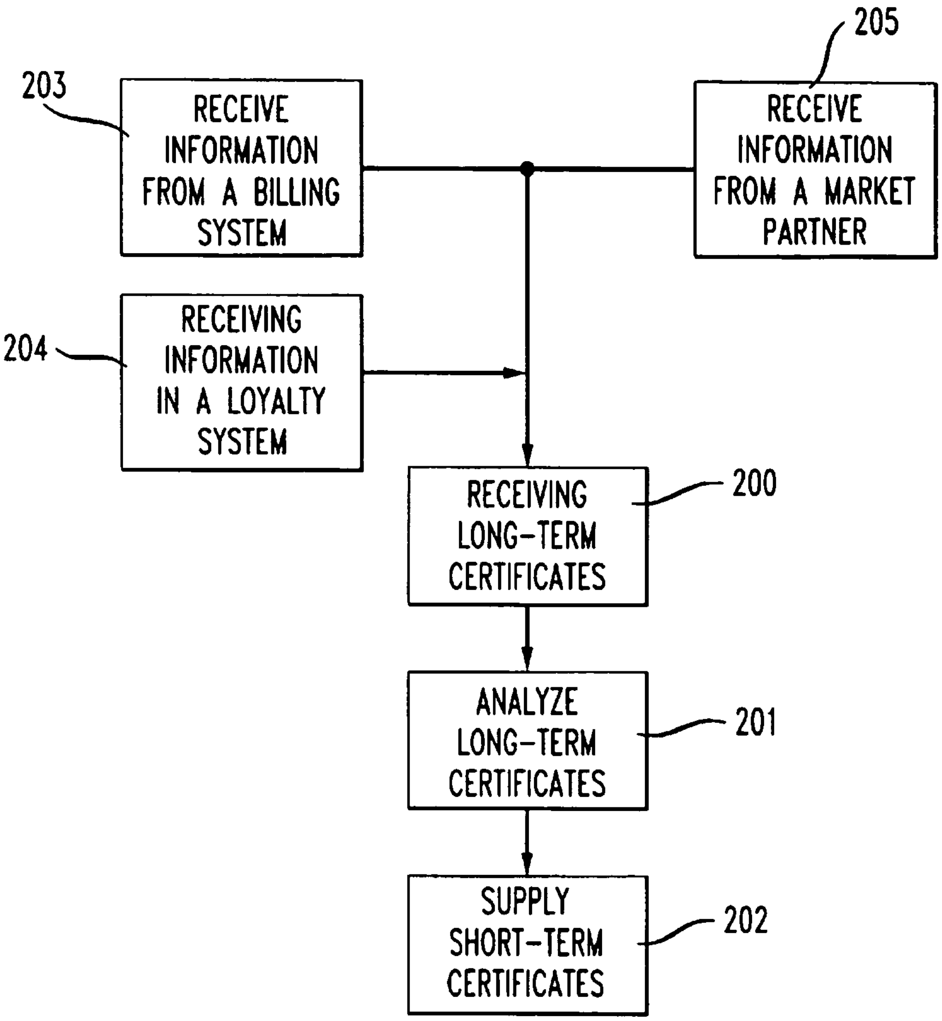
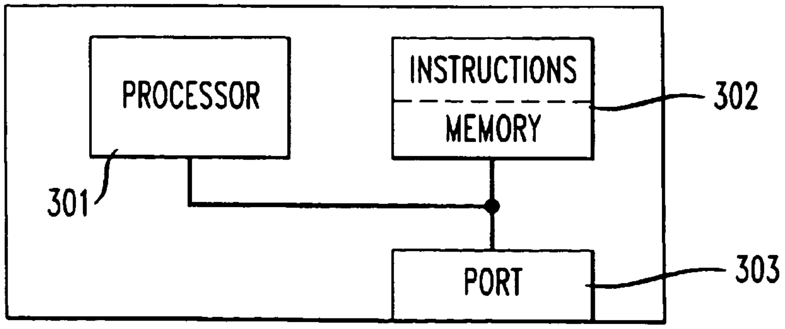

205
203
RECEIVE INFORMATION FROM A BILLING SYSTEM
RECEIVE INFORMATION FROM A MARKET PARTNER
204
RECEIVING INFORMATION IN A LOYALTY SYSTEM
RECEIVING LONG-TERM CERTIFICATES
200
ANALYZE LONG-TERM CERTIFICATES
201
SUPPLY SHORT-TERM CERTIFICATES
202

FIG. 3


PROCESSOR
INSTRUCTIONS
MEMORY
302
301
PORT
303

METHOD AND APPARATUS USING DIGITAL CREDENTIALS AND OTHER ELECTRONIC CERTIFICATES FOR ELECTRONIC TRANSACTIONS

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority to Provisional application Ser. No. 60/060,643, filed on Oct. 1, 1997.

FIELD OF THE INVENTION

The present invention relates to digital credentials and other electronic certificates. More particularly, the present invention relates to a service for using digital credentials and other electronic certificates to practice commerce on a network.

BACKGROUND OF THE INVENTION

To exercise certain rights and privileges, people need to possess or show various types of credentials. Credentials are certificates such as birth certificates, Social Security Cards, driver's licenses, membership cards, admission badges charge cards, and the like that represent some certified assertion about a person. In the case of a driver's license, an officer of the state certifies that a specific person is licensed to drive a vehicle. A charge card represents an assertion, certified by some bank or other organization, that a person has a charge account at that bank. Companies issue credentials for their employees, usually in the form of ID badges. Generally. the certificate will include some means of identifying to whom the assertion applies (the holder or subject of the credential), and who is certifying the assertion (the certifier of the credential, who is often the issuer).

In the case of a driver's license or corporate ID, the holder is typically identified by a photograph and signature specimen laminated to the certificate and the certifier of the credential is usually identified by a logo, layout, and some other means such as a hologram.

With the advent of electronic commerce, standard credentials have become insufficient, and the need for digital credentials has become more widespread. Digital credentials are electronic certificates having the property that the assertions about the holder can be interpreted and verified by a computer, the certifier can be reliably recognized by a computer, and the holder's present intention to use the credentials can be recognized by a computer (often remotely, through a network). Digital credentials can use a cryptographic mechanism known as a digital signature. An electronic document can be signed by applying a cryptographic secret key controlled by the signer. A signature can be verified using public information (known as the public key). The verification process can use the public key to verify that the signer's secret key was used to sign the document. The science of public key cryptography enables this.

Examples of digital credentials are automatic teller machine (ATM) or bank cards. As opposed to other types of certificates mentioned earlier, these are not usually presented to people for verification. They are normally presented to an ATM and ultimately to a specialized computer network. The relevant information regarding the certifier is digitally encoded on a magnetic strip and the cardholder is identified by a Personal Identity Number or PIN. Furthermore, the holder's present intention to apply the rights asserted by the credential (such as withdrawing money) is signified by the holder's entry of the PIN. This ATM card allows the holder to use electronic banking over specialized digital networks. The present form of digital credentials, however, can support only a minimal variety of services over specialized and non-specialized networks such as the Internet.

Present ways of using digital credentials (using PINs and passwords) are notoriously insecure, very user-unfriendly, and generally inadequate for electronic commerce. For example, while hand-written signatures on documents can make paper records auditable, PINs and passwords are not very useful for this purpose. In particular, they do not have persistent properties as signatures do. For example, one can directly verify a signature post-hoc, but PINs and passwords can be verified only at time of use. The certified digital signature can substitute for a hand-written signature.

The importance of digital credentials is rapidly increasing because networks are becoming more open and public. Whereas a person's identity on a closed network is known through a network operating system, and privileges can be determined by database look-ups, such is not the case on the Internet, for example.

Digitally-signed certificates have been used in electronic payment systems that have arisen over the past five years or so. At least three distinct types of payment systems exist, each of which differs from the current invention in significant ways. The three systems are referred to as e-check, e-charge, and e-cash.

An e-check is designed to function in a way similar to the way paper checks function. While a paper check is a signed request for a bank to pay a given amount from the payer's account to the party that is named on the check (the payee), an e-check is a message requesting the same procedure, but it is electronically signed by the payer. The electronic signature certifies, as in the case of a paper check that the user attests to the payment request and to the specifics of the payee and the amount. With a paper check, the payee has the option of verifying the identity of the payer in person, often demanding one or more alternate methods of payer identification, or the payee can sometimes wait until the check "clears" before providing value in return for the check. Clearing means that the payee's bank receives payment from the payer's bank. With an e-check system, the payee can also wait until the check clears from the payer's bank, or the payee can accept the legitimacy of the payer's digital signature by checking the certificate that the payer's bank issues to the payer which certifies the payer's signing key. In the latter case, the payee risks the possibility that the digital signature certificate has been revoked. This risk is reduced when the payee checks an electronic "Certificate Revocation List" or CRL. Nonetheless, the residual risk exists that the CRL is not up to date. Additionally, the traditional risk exists that the payer's account may have insufficient funds, and the e-check will not clear.

E-checks use the same clearing system and clearing networks used by paper checks. The systems and networks are relatively expensive to use, and when one adds the cost of administering CRLs and the cost of processing e-checks returned for insufficient funds, the use of e-checks for relatively small payments of a few dollars or less is not cost effective. In the present invention, these inefficiencies are addressed by reducing the dependency on CRLs, and by use of a novel approach to risk management, integrating risk management parameters directly into a certificate.

Another use of digital certificates in payment systems is illustrated by the Secure Electronic Transaction ("SET") standard that has been proposed by MasterCard and Visa. SET describes a relatively complex mechanism for making a payment using certificates within the current credit card payment support infrastructure. A number of parties exist in SET: the cardholder, the payee (or merchant), the issuing bank, the acquirer (or merchant's bank), the payment gateway, and optionally, "third parties" that represent one or more of the financial institutions involved. In SET, five different parties have certificates. Cardholder certificates function as an electronic representation of the payment card. Merchant certificates function as an electronic substitute for the payment brand decal that appears in a store window. Payment Gateway certificates are used by Acquirer's or their processors for the systems that process authorization and capture messages. In addition, Acquirer certificates and Issuer certificates aid in the distribution of Merchant and Issuer certificates, respectively. In general, the various certificates are used to support cryptographic keys that are used to provide credit card transaction messages with security properties such as privacy and authenticity.

SET is, overall, an elaborate scheme that is described in the "SET Secure Electronic Payment Transaction Specification" published by MasterCard and Visa. The certificates involved in SET may need to be revoked for any of a number of reasons such as key compromise, or change of status of the party holding the certificate. In contrast to the present invention, the scheme requires a certificate hierarchy, on-line verification procedures, and a certificate revocation infrastructure. Transactions require a significant amount of computation by multiple parties to complete.

Another use of digital certificates in payment systems is illustrated in electronic cash (e-cash) systems where cash is either represented by digital bearer certificates or by "value registers" in smart cards. In the case of digital bearer certificates, a digital signature is applied to an assertion that the certificate may be redeemed for a certain amount of cash at a certain bank or financial institution. A bank will issue certificates that can be used to verify the authenticity of the signature on the bearer certificate. Because digital bearer certificates can be freely copied, a risk exists that users will attempt to repeatedly use the same certificate. Therefore, risk management measures must be employed to ensure that each certificate is spent precisely once. Typically, either a smart card is used to contain the certificates and to participate in a two party protocol that marks certificates as used, or a network-based mechanism may be employed that records each certificate as it is used, and allows any payee to see if the certificate tendered is being used for the first time.

In the case of value registers in smart cards, certificates are used to certify the keys used to verify the digital signatures on messages that are exchanged between two software applications running on the smart cards. For example, a payer's smart card debits its value register (or current cash balance), and signs and sends a message affirming the act to the payee. The payee, upon receiving the message affirming the debit can check the signature on the certificate and verify the signature on the message.

Multiple risks exist in this system as well. In particular, the credit and debit operations must be encapsulated within smart cards or some other physically secure containers that must be distributed and maintained. In addition, should the certificates be compromised, counterfeit e-cash can be produced that is indistinguishable from e-cash that is issued by a legitimate originator. Should the physical container of a card be compromised, then clones of that card could be created that never debit their balances but nonetheless dispense e-cash acceptable to other cards. These are called "golden goose" cards. Thus, this type of e-cash, as a payment system, requires significant risk management measures. Another difficulty associated with this payment scheme has to do with recovery from errors. A communication error can literally destroy value. For example, if one smart card sends a signed message "I have debited my value register by $20" to another smart card, yet the second smart card does not receive that message intact, no credit will be offset to the debit. A support structure to make amends for these type of errors is required.

The shortcomings with the prior art involve the difficulty in using credentials that have been distributed electronically in a highly distributed system that lacks a reasonable means to revoke or update the credentials. For example, assume one holds a digital credential that authorizes the holder to purchase goods up to a value of one hundred thousand dollars (e.g., a corporate credit card). To use this credential, one must go to a central database to re-verify each time the credential is used.

Within the known systems, risk management measures are required to properly support payment systems, and defend them against fraud. Yet the known systems do not contain an efficient way in which risk management is integrated into the payment system.

SUMMARY OF THE INVENTION

The present invention relates to a method and apparatus for using digital credentials, or certificates to facilitate commerce on a network. In one embodiment of this invention, a party wishing to act as guarantor of a transaction would receive long-term certificates from a consumer after the consumer logs into the network. The guarantor analyzes the long-term certificates, at least to verify the identity of the consumer. The guarantor, after being satisfied with the information presented, supplies short-term certificates containing assertions based on information from the above analyses. The short-term certificates can then be used to purchase goods from participating merchants on a network.

In another embodiment, merchants use the short-term certificates to verify terms and conditions under which a given consumer can be billed through the guarantor. The short-term certificates also certify the cryptographic public keys of consumers that are used to digitally sign statements requesting merchants to bill for goods and services purchased through the guarantor. Billing records associated with purchases are forwarded to the guarantor or his agent, whereby the records are sorted by consumer identity and used to construct periodic statements containing many billing records that are made available to consumers who can make a single payment. Detailed information about the purchases is thus provided to the guarantor who then helps merchants market goods accordingly. The billing records may contain digitally signed statements by consumers directing the merchant to bill through the guarantor.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a system-level block diagram of an embodiment of the present invention.

FIG. 2 is a flow chart of an embodiment of a method of the present invention.

FIG. 3 illustrates an embodiment of an apparatus and system in accordance with the present invention.

DETAILED DESCRIPTION

The present invention is directed to ways of using digital credentials and other electronic certificates to practice commerce over a network. The purpose is to run a relatively convenient and efficient system using a combination of both long-term and short-term certificates.

Long-term certificates, as defined here, are certificates that contain information or make an assertion that is not expected to change over some long period of time. For example, long-term certificates can be used to represent a person's identity. Revocation of long-term certificates is not necessary on any large scale because the information contained in long-term certificates is relatively static and benign.

Short-term certificates, on the other hand, hold information or make assertions that may rapidly change, and therefore are designed to expire after some relatively short period of time. For example, short-term certificates may contain information about a person's credit history, shopping history, or information about the short-term certificate's maximum value as currency. Short-term certificates may make assertions about what a person is authorized to do, or about agreements that they may have with other parties.

The validity of the short-term credentials can be based on an individual's identity. For example, when a person logs into a system, the person uses some means to verify identity (using long-term credentials, for example), and then the system supplies short-term credentials which say, for example, that the client is authorized to charge for commerce on the world wide web for purchases the amount of which is not to exceed some fixed amount. Typically the short-term credential can also certify cryptographic keys that can be used for digital signatures that affirm a person's agreement with a contract. In addition, the short-term credential might contain the semantics attributed to the use of the person's digital signature as well as statements of limitations of liability.

Referring now in detail to the drawings, FIG. 1 illustrates a system-level embodiment of the present invention. In this system, Customer Client **104** desires to purchase goods or services from Merchant **105**. To do this, Customer Client **104** needs to present to Merchant **105** a form of payment that will be accepted by Merchant **105**. In anticipation of this, Client **104** may present a long-term certificate to a certifier to access a certificate of payment called a short-term certificate.

The long-term certificate can be certified through known encryption techniques. The certifier is typically, for example, an internet service provider, bank, or any entity designed to certify credentials. The long-term certificate contains, at the very least, information that verifies the identity of Customer Client **104**. The long-term certificate may contain other information desired by the certifier. Once the certifier is satisfied by Customer Client **104**'s long-term certificate information, the certifier sends Customer Client **104** one or more short-term certificates from the short-term certificate database **103**.

Short-term certificates are digital in form, and contain information stating, at least, that the certifier guaranties payment up to a certain amount of value. In addition, the short-term certificate can contain marketing information. For example, a short-term certificate can tell a participating merchant that goods and services may be charged by the client named in the certificate to a specific account, through an agreed-upon channel, for up to the amount of $20. In addition, the short-term certificate may contain information that instructs Merchant **105** to apply a 20% discount to the cost of the goods supplied to the bearer of this short-term certificate. Upon receiving a short-term certificate, Merchant **105** can send an optional query to the short-term database for various reasons such as double-checking the certificate's validity in the case when the purchase amount exceeds some threshold stated in the short-term certificate. The short-term certificates are short term in the sense that they contain information or make assertions based on information that may change over a relatively short period of time. They therefore can be set to expire in some short period of time. For example, a certifier may supply a short-term certificate to Customer Client **104** that guaranties that the Client can charge to an account the purchase of any item that costs up to $20, but can only be used within 24 hours after Customer Client **104** receives this certificate.

Merchant **105** and Customer Client **104** consummate a transaction by promising (on the part of Merchant **105**) to supply goods or services in exchange for an affirmative indication on the part of the Client that the goods or services can be charged to a billing account maintained in Billing System **106** according to and limited by the information provided by a short-term certificate. Once the short-term certificate is received, and the transaction is completed, the short-term certificate is sent along with an electronic record of a bill of sale through agreed-upon channels for payment from the certifier, or guarantor.

The above-mentioned agreed-upon channels, called Billing System **106**, collect billing records, and their corresponding short-term certificates and renders them for payment. In addition to serving as a conduit for payment, the billing system may supply information to various subsystems that serve to analyze information about the transaction. The Transaction Analysis **107** collects details of the transaction. The Transaction Analysis **107** correlates different types of purchases with different demographics of this particular Customer Client **104**, and then determines what offers might be made to this particular consumer. The purpose of the transaction analysis is to determine patterns of consumer behavior so that some action may be taken. For example, Customer Client **104** might show a pattern of behavior that would alert the certifier that Customer Client **104** is in the market for an automobile. In other words, transactional information is used to better match marketing with consumer-behavior information.

Once the transactional analysis is complete, the results are used in Offer Management **102** to market goods or services to Customer Client **104**, possibly by attaching offers to short-term certificates in Short-Term-Certificate Database **103**. In this way, a type of high-gain feedback loop is completed, as can be seen in FIG. 1.

In FIG. 1, Offer Management **102** can use information received by Risk Management System **102**(*a*), Loyalty System **102**(*b*), and Market Partners **102**(*c*) to determine what, if any, information should go into the short-term certificates along with any assertions that might be made about terms and conditions, credit limits, discounts, etc. Risk Management System **102**(*a*) can receive information from Billing System **106**, thereby keeping data on a particular Customer Client's usage patterns. Risk Management System **106** can then analyze the information supplied by Billing System **106**, and alert the certifier as to how much risk should be taken with regard to a particular Customer Client. For example, Risk Management System **102**(*a*) can alert the certifier to change the credit limit, either up or down, for a particular Customer Client. The system also can determine whether or not the recent usage patterns of a person are indicative of fraud or other misuse (that may have resulted from a key management compromise whereby a consumer's identity certificate and secret key have been compromised).

This information passed between Billing System **106**, Risk Management System **102**(*a*), and the certifier can be updated and analyzed arbitrarily quickly, possibly on a daily basis. This rapid response obviates the need for use of certificate revocation lists.

Billing System **106** can also supply information to Loyalty System **102**(*b*). Loyalty System **102**(*b*) is a system whereby consumers are rewarded for regular use of a particular merchant. An example of a loyalty system is found in frequent-flier programs. Loyalty System **102**(*b*) can collect and analyze information, and then supply this information to the certifier's Offer Management **102** so the certifier can tailor its marketing through Offer Management **102** accordingly. In particular, the Offer Management process can author assertions to be inserted into the short-term certificates that declare that loyalty points are available to pay for purchases from participating merchants. Such a merchant can thus accept payment ostensibly in loyalty points, but the merchant can be remunerated through the billing system in cash or other consideration upon presentment of a certificate-backed, signed purchase agreement. This system offers an advantage over other loyalty systems because one purpose of a loyalty system is to reinforce good behavior by rewarding the user, and this system can reward the user arbitrarily rapidly.

Market Partners **102**(*c*) can enter into agreements with certifiers to help the certifier tailor its marketing through Offer Management **102**. The idea is to capture the value of transactional information without severely impacting the consumer's privacy. Market Partner **102**(*c*) provide information to the system about what Market Partner **102**(*c*) desires in a consumer. This information might be a demographic profile, a consumer-behavior profile, etc. For example, Market Partner **102**(*c*) can tell the certifier that it wishes to target people who are shopping for new cars. Offer Management **102** then correlates the needs of Market Partner **102**(*c*) with the information it contains about the consumer.

FIG. **2** is a flow chart of a process in accordance with an embodiment of the present invention. In its most basic form, long-term certificates, or some other proof of identity are received by the certifier at step **200**. At step **201**, the certifier then analyzes the information presented in the long-term certificate and then, at step **202**, supplies, from a short-term-certificate database, short-term certificates that can be used as instruments to purchase goods from merchants on the network.

In addition to receiving long-term certificates, the certifier may receive, at step **203** information from a billing system, at step **204** information from a market partner, and at step **205** information from a loyalty system.

The short-term certificate can contain a maximum value for which certifier will act as guarantor upon presentment by a merchant. In addition, the short-term certificate can contain information about offers to the consumer, incentive programs, or loyalty programs.

As stated above, various subsystems, such as a risk management system, a loyalty system, or a marketing system can be interposed between the certifier and the merchant. The short-term certificate can contain information reflecting, for example, the risk-management analysis with regard to a consumer, the loyalty-system analysis with regard to a consumer, or the marketing analysis with regard to the consumer. For example, the short-term certificate can contain a limit on the certificate's guaranty limits based on the risk-management analysis; the certificate can contain a number of acquired consumer points based on the loyalty-system analysis; and the certificate can contain offers (including incentives) to the consumer based on the marketing analysis.

When a consumer desires to make a purchase from a participating merchant, he or she presents through the network one or more short-term certificates. The merchant can analyze the short-term certificate, and determine any guarantees of payment, any rights to use alternative methods of payment such as loyalty points, any discounts or other entitlements, and then make appropriate adjustments to the consumer's bill of sale. The merchant's final price, terms, and conditions for a sale as part of a bill-of-sale, are forwarded to the consumer, who will indicate acceptance, and make the purchase through some affirmative act (that may be required by a condition stated in the short-term certificate) such as signing the bill of sale with a digital signature whose verification key is certified by the short-term certificate.

Ultimately, the certifier can collect for the goods or services furnished guaranteed by creating a billing record containing references to sending the bill of sale and the short-term certificate obtained from the user, and forwarding this billing record through a regular billing channel to the certifier. The certifier can then collect all billing records associated with a specific user and present them to the user in a statement. For example, if the certifier is a telephone company, the telephone company can bill the user for amounts as stated in the short-term certificate by using the user's regular monthly telephone bill.

FIG. **3** shows an embodiment of an apparatus in accordance with the present invention. The apparatus includes processor **301**, memory **302** that stores instructions adapted to be executed by processor **301**, and port **303** adapted to be connected to a network, with both port **303** and memory **302** coupled to processor **301**. Memory includes any medium capable of storing instructions adapted to be executed by a processor. Some examples of such media include, but are not limited to, floppy disks, CDROM, magnetic tape, semiconductor memory, hard drives, and any other device that can store digital information. In one embodiment, the instructions are stored on the medium in a compressed and/or encrypted format. As used herein, the phrase "adapted to be executed by a processor" is meant to encompass instructions stored in a compressed and/or encrypted format, as well as instructions that have to be compiled or installed by an installer before being executed by the processor.

In one embodiment of the present invention, memory **302** stores instructions adapted to be run on processor **301**, to receive information, analyze that information, and then supply short-term certificates the character of which depends on the results of the analysis. The information received and analyzed can come from market partners, a billing system, a loyalty system, and from long-term certificates supplied by a consumer.

As explained in detail above the invention increases efficiency and productivity of commerce on a network. By using digital credentials and other digital certificates, micro-billing becomes more feasible by decreasing transaction costs, limiting risk, and allowing for easily updated credentials.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

What is claimed is:

1. A method for performing an electronic transaction, comprising:

(a) receiving a long-term certificate;

(b) authenticating a user associated with the long-term certificate;

(c) sending a short-term certificate to the user authenticated in (b), the short-term certificate containing authorization to perform commerce over a network.

2. The method of claim 1, further comprising:

(d) evaluating a risk associated with the user; and

(e) including in the short-term certificate information about the risk associated with the user.

3. The method of claim 2, wherein the risk associated with the user is reflected in an upper limit on the short-term certificate's value.

4. The method of claim 1, further comprising:

(d) receiving information about the user's spending history; and

(e) including in the short-term certificate information based on the user's spending history.

5. The method of claim 4, wherein the information about a user's spending history includes marketing offers.

6. The method of claim 1, further comprising:

(d) receiving from a market partner information about the market partner's needs; and

(e) including in the short-term certificate information about the market partner's needs.

7. The method of claim 6, wherein the information about a market partner's needs includes marketing offers.

8. The method of claim 2, further comprising:

(f) receiving, information about the user's spending habits; and

(g) including in the short-term certificate information about the user's spending habits.

9. The method of claim 2, further comprising:

(f) receiving from a market partner information about the market partner's needs; and

(g) including in the short-term certificate information about the market partner's needs.

10. The method of claim 4, further comprising:

(f) receiving from a market partner information about the market partner's needs; and

(g) including in the short-term certificate information about the market partner's needs.

11. The method of claim 8, further comprising:

(h) receiving from a market partner information about the market partner's needs; and

(i) including in the short-term certificate information about the market partner's needs.

12. The method of claim 1, further comprising:

(d) billing the user through a regular billing channel between the certifier and the user.

13. The method of claim 12, wherein the regular billing channel is a telephone bill.

14. The method of claim 12, wherein the regular billing channel is a credit-card bill.

15. The method of claim 8, further comprising:

(h) billing the user through a regular billing channel between the certifier and the user.

16. The method of claim 11, further comprising:

(j) billing the user through a regular billing channel between the certifier and the user.

17. An apparatus for practicing commerce on a network, comprising:

(a) a processor;

(b) a port coupled to said processor; and

(c) a memory, also coupled to said processor, storing instructions adapted to be executed by said processor to

(i) receive a long-term certificate;

(ii) authenticate a user associated with the long-term certificate; and

(iii) send short-term certificates to the user authenticated in (ii), the short-term certificate containing authorization to perform commerce over a network.

18. The apparatus of claim 17, further comprising:

(d) a memory storing instructions adapted to be executed by said processor to

(i) evaluate the risk associated with the user; and

(ii) include in the short-term certificate information about the risk associated with the user.

19. The apparatus of claim 18, wherein the risk associated with the user is reflected in an upper limit on a value of the short-term certificate.

20. The apparatus of claim 17, further comprising:

(d) a memory storing instructions adapted to be executed by said processor to

(i) receive information about the user's spending history; and

(ii) include in the short-term certificate information based on the user's spending history.

21. The apparatus in claim 20, wherein the information about the user's spending habits includes marketing offers.

22. The apparatus of claim 17, further comprising:

(d) a memory storing instructions adapted to be executed by said processor to

(i) receive from a market partner information about the market partner's needs; and

(ii) include in the short-term certificate information about the market partner's needs.

23. The apparatus of claim 22, wherein the information about the market partner's needs includes marketing offers.

24. The apparatus of claim 18, further comprising:

(e) a memory storing instructions adapted to be executed by said processor to

(i) receive information about the user's spending habits; and

(ii) include in the short-term certificate information about the user's spending habits.

25. The apparatus of claim 18, further comprising:

(e) a memory storing instructions adapted to be executed by said processor to

(i) receive from a market partner information about the market partner's needs; and

(ii) include in the short-term certificate information about the market partner's needs.

26. The apparatus of claim 20, further comprising:

(e) a memory storing instructions adapted to be executed by said processor to

(i) receive from a market partner information about the market partner's needs; and

(ii) include in the short-term certificate information about the market partner's needs.

27. The apparatus of claim 24, further comprising:

(e) a memory storing instructions adapted to be executed by said processor to

(i) receive from a market partner information about the market partner's needs; and

(ii) include in the short-term certificate information about the market partner's needs.

28. A computer-readable medium that stores instructions adapted to be executed by a processor to perform the steps of:

(a) receiving a long-term certificate;

(b) authenticating a user associated with the long-term certificate;

(c) sending a short-term certificate to the user authenticated in (b), the short-term certificate containing authorization to perform commerce over a network.

29. The computer-readable medium of claim 28, further comprising

(d) evaluating the risk associated with the user; and

(e) including in the short-term certificate information about the risk associated with the user.

30. The computer-readable medium of claim 28, further comprising:

(d) receiving information about the user's spending history; and

(e) including in the short-term certificate information about the user's spending history.

31. The computer-readable medium of claim 28, further comprising:

(d) receiving from a market partner information about the market partner's needs'

(e) including in the short-term certificate information about the market partner's needs.

* * * * *